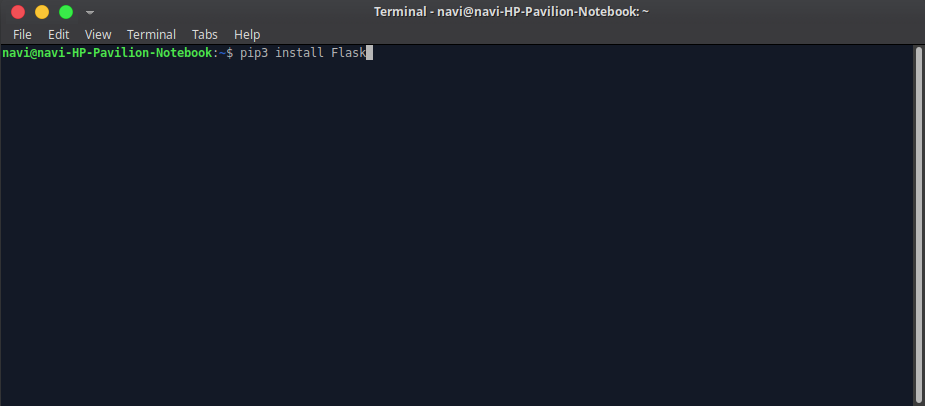
# Practica de servicios y contenedores

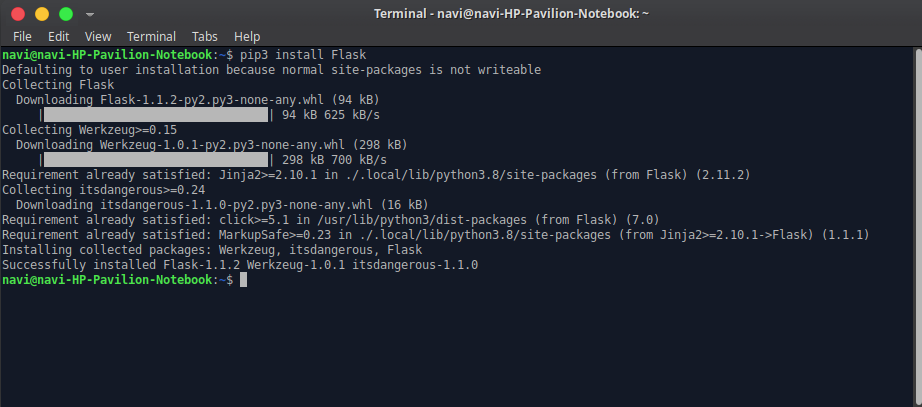
Iván Castañeda

Se debe contar con los siguientes paquetes instalados

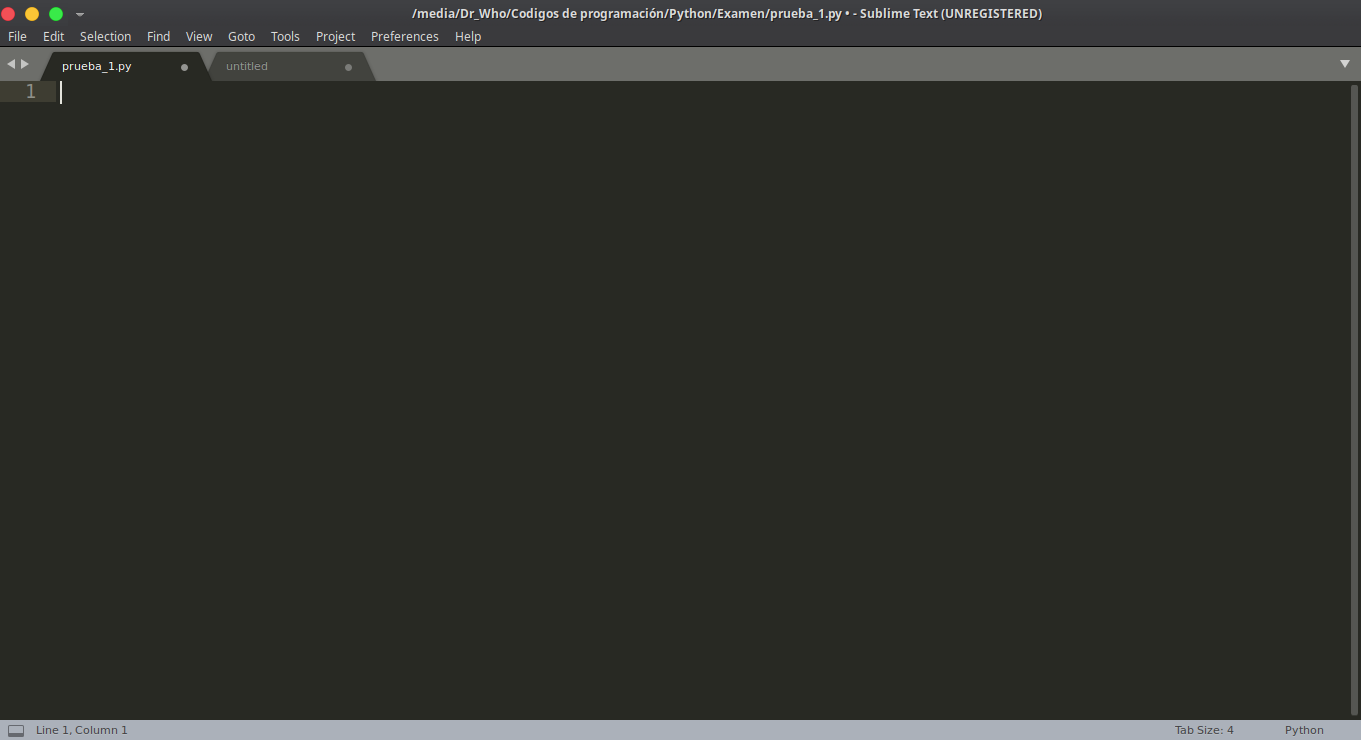
Flask

Instalando el paquete flask con pip3





Una vez instalado flask, se trabaja con el IDE Sublime Text 3



De la biblioteca flask se importan los metodos Flask que es un framework de python, junto con jsonify para mostrar los mensajes en formato Json from flask import Flask, jsonify

Se crea un objeto de la clase app = Flask(\_\_name\_\_)

Se utiliza @app.route() para indicarle a Flask que url debe activar @app.route('/api/sps/helloworld/v1') por default el método que inicia es [‘Get’], aunque se pueden añadir más

if \_\_name\_\_ == '\_\_main\_\_' → \_\_name\_\_ es una variable especial en python, asegura que nuestra aplicación Flask se ejecute solo cuando se ejecuta el archivo principal y no al importar otro archivo

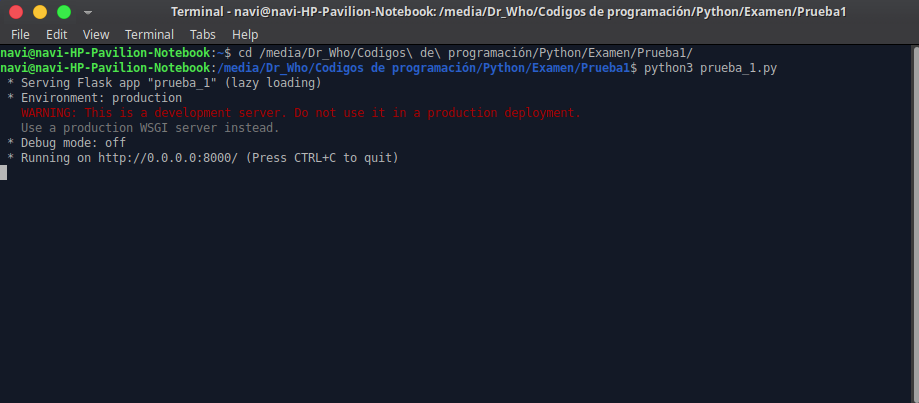
app.run(host='0.0.0.0', port=8090) ejecuta la aplicación flask



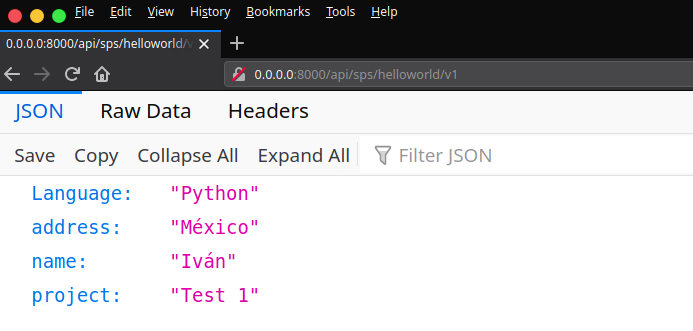
Métodos:

* /
  + Envia un mensaje de Hello World!
* <int:number>
  + Incremente en 1 el numero escrito
* <string:name>
  + Manda saludo personalizado al escribir el nombre
* /api/sps/helloworld/v1'
  + Envia un archivo json con el diccionario mostrado en la imagen anterior

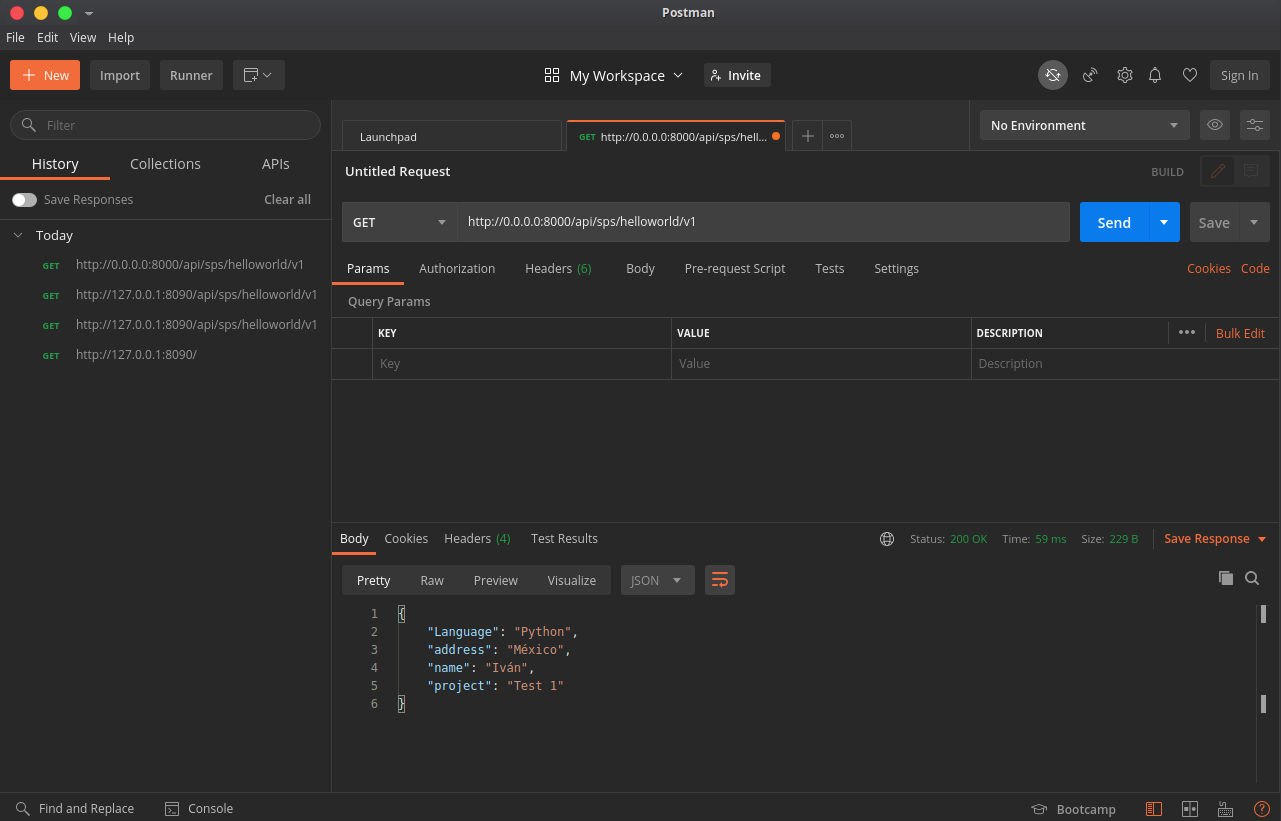
Se ejecuta la aplicación en la terminal de la forma python3 prueba\_1.py



Inicia el servidor web y al acceder a la dirección del endpoint, se muestra lo siguiente



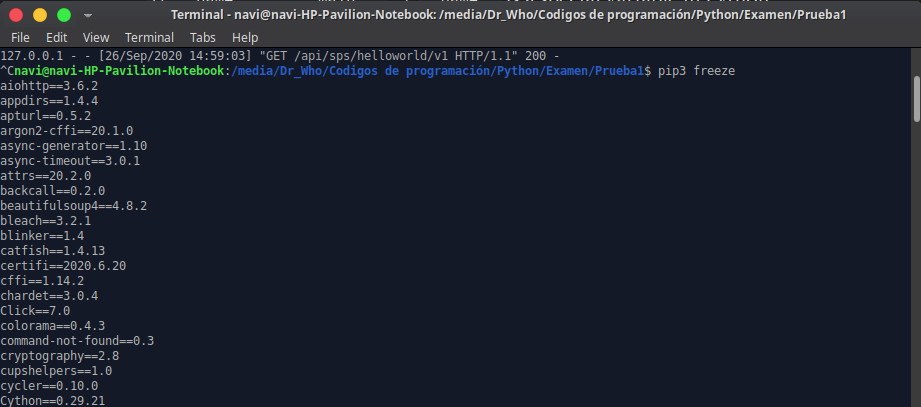
Se utiliza postman para confirmer que todo se muestra de forma adecuada



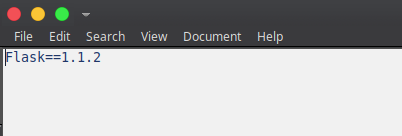
Se instala docker (en mi caso sera en ubuntu) a partir de la siguiente guia https://docs.docker.com/engine/install/ubuntu/

(No tome capturas de la instalación de docker)

Mientras se instala docker, se ejecutará el comando pip3 freeze el cual índica los paquetes que se han instalado



Se ejecuta el comando pip3 freeze > requirements.txt, el commando crea un archivo con extensión txt que contiene todos los paquetes instalados en el pc, de este archive creado solo se deja flask



Una vez instalado docker, se crea un archivo llamado dockerfile, el cual tendra las instrucciones para crear el contenedor

FROM alpine:3.10 sistema operativo del contenedor

RUN apk add --no-cache python3-dev \

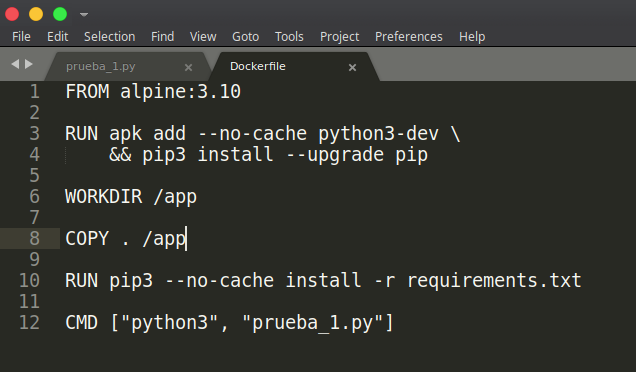
&& pip3 install --upgrade pip instalasión de python y pip3

WORKDIR /app el directorio de trabajo

COPY . /app copia los archivos en el directorio de trabajo

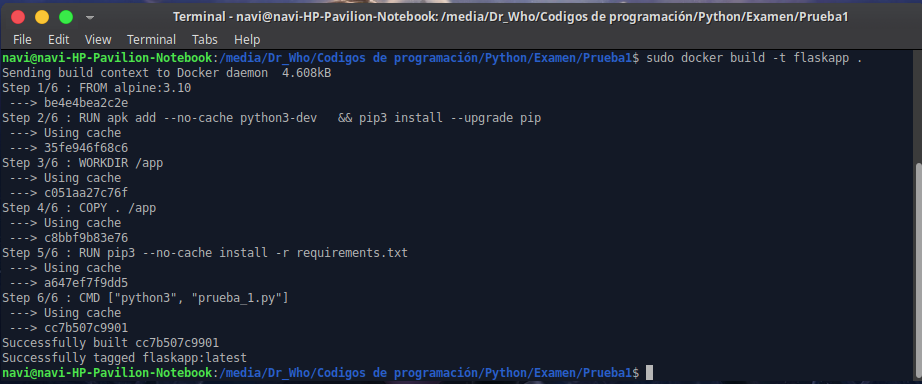
RUN pip3 --no-cache install -r requirements.txt instala los paquetes a utilizar del archivo

CMD ["python3", "prueba\_1.py"] ejecuta los comandos para iniciar la prueba



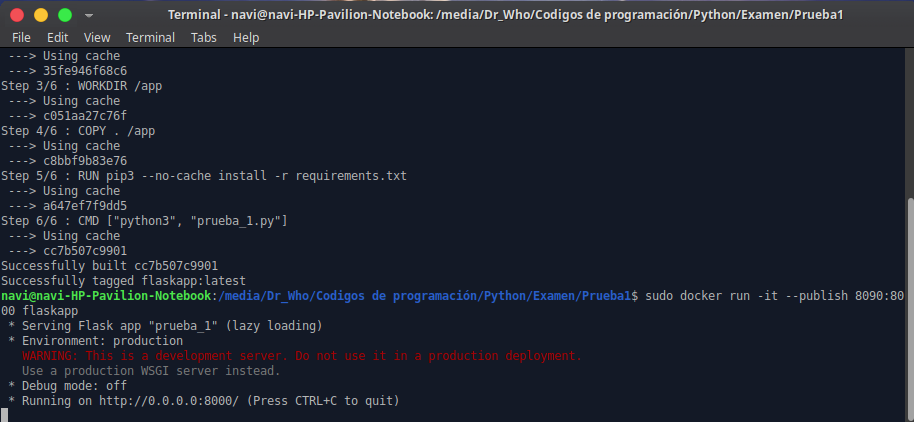
En nuestra terminal se ejecutan los siguientes comandos

sudo docker build -t flaskapp .



El comando sirve para crear el contenedor, seguidamente se ejecuta el siguiente comando

sudo docker run -it --publish 8090:8000 flaskapp



En donde el localhost a consultar sera 8090, con el navedor se ingresa y listo

